# Symmetry of information and nonuniform lower bounds

Sylvain Perifel

# Symmetry of information and nonuniform lower bounds

Sylvain Perifel

LIP*, École Normale Supérieure de Lyon.
Sylvain.Perifel@ens-lyon.fr

**Abstract.** In the first part we provide an elementary proof of the result of Homer and Mocas [3] that for all constant $c$, the class $\mathsf{EXP}$ is not included in $\mathsf{P}/n^c$. The proof is based on a simple diagonalization, whereas it uses resource-bounded Kolmogorov complexity in [3].
In the second part, we investigate links between resource-bounded Kolmogorov complexity and nonuniform classes in computational complexity. Assuming a weak version of polynomial-time symmetry of information, we show that exponential-time problems do not have polynomial-size circuits (in symbols, $\mathsf{EXP} \not\subset \mathsf{P/poly}$).
*Keywords:* computational complexity, nonuniform lower bounds, resource-bounded Kolmogorov complexity, symmetry of information

## 1 Introduction

Whereas some uniform lower bounds have been proved long ago thanks to hierarchy theorems, little progress have been made towards longstanding open problems concerning nonuniform lower bounds in complexity theory. This observation is explained by the lack of proof techniques for these nonrelativizing questions. In particular, a simple diagonalization is not suitable for the main question of whether exponential-time problems have polynomial-size circuits. Yet the advantage of diagonalization is its simplicity. In this paper, we show that a hypothesis of resource-bounded Kolmogorov complexity gives diagonalization enough power to settle these questions.

We are interested in the question of whether the class $\mathsf{EXP}$ of problems decided in exponential time has polynomial-size circuits (in symbols, whether $\mathsf{EXP} \subset \mathsf{P/poly}$). As mentioned above, the separation $\mathsf{EXP} \neq \mathsf{P}$ is well-known but this nonuniform counterpart is still open. On this problem, two approaches have yielded significant results.

The first approach was to find the smallest uniform class provably not contained in $\mathsf{P/poly}$. In this direction, Kannan [4] proved that $\mathsf{NEXP}^{\mathsf{NP}}$ does not have polynomial-size circuits, and afterwards Schöning [12] gave a simplified proof that $\mathsf{EXPSPACE}$ does not have polynomial-size circuits. Here, we see that performing a diagonalization out of $\mathsf{P/poly}$ requires more than exponential time. Later, the second approach was to obtain the best nonuniform lower bound for

* UMR 5668 ENS Lyon, CNRS, UCBL, INRIA. Research report RR2006-50.

EXP problems. Homer and Mocas [3] showed that EXP does not have circuits of size $n^c$ for any fixed constant $c$.

As a first step toward our main theorem, another proof of this last result is provided in the first part of the present paper (Proposition 2). This is an elementary proof consisting in a mere diagonalization (whereas the original proof of [3] makes use of resource-bounded Kolmogorov complexity) which is included here because it familiarizes with the proof of the main result of the paper, and also because this easy proof has never been published to the author's knowledge (though it uses techniques very similar to [12]). As a corollary, included here as another illustration of this method, we obtain a nonuniform lower bound on PP problems (Proposition 4). Unfortunately this is much weaker than the result of Vinodchandran [13].

In the second and main part, we show that an assumption of resource-bounded Kolmogorov complexity enables to combine both approaches described above. Namely, if a weak version of polynomial-time symmetry of information holds true, then $\mathsf{EXP} \not\subset \mathsf{P/poly}$ (Theorem 2). This result therefore relates two major open questions. The proof once again consists in a simple diagonalization.

Symmetry of information is a beautiful theorem in Kolmogorov complexity and one of its versions can roughly be stated as follows: if $x$ and $y$ are two words, $x$ contains the same quantity of information on $y$ as $y$ on $x$. This theorem is due to Levin [14] and Kolmogorov [6].

When requiring polynomial time bounds on the computations, however, the similar property, called polynomial-time symmetry of information, is a challenging open problem in resource-bounded Kolmogorov complexity. This problem has already been related to computational complexity by at least two results. First, Longpré and Watanabe [9] show that if $\mathsf{P} = \mathsf{NP}$ then polynomial-time symmetry of information holds. Second, more recently and closer to our present preoccupations, Lee and Romashchenko [7] show that if polynomial-time symmetry of information holds, then $\mathsf{EXP} \neq \mathsf{BPP}$. A longer discussion on symmetry of information, inspired by the introduction of [7], is provided in Section 4.

Here, assuming a weak version of polynomial-time symmetry of information (see Section 4) we prove that $\mathsf{EXP} \not\subset \mathsf{P/poly}$ (a stronger conclusion than in [7] since $\mathsf{BPP} \subset \mathsf{P/poly}$, see [1]). As we shall see, symmetry of information enables to divide advices into small blocks on which diagonalization can be performed in EXP.

All these results teach us that polynomial-time symmetry of information, even in its weakest forms, is a hard but central question to study. Indeed, if it holds, then EXP does not have polynomial-size circuits, else $\mathsf{P} \neq \mathsf{NP}$. In both cases, a fundamental question in complexity theory would find an answer.

*Organization of the paper.* Section 2 is devoted to definitions and notations in computational complexity and resource-bounded Kolmogorov complexity. Section 3 consists of another proof of the result of [3] that exponential-time problems do not have circuits of any fixed polynomial size $n^c$. A simple corollary is also shown there, namely a nonuniform lower bound on PP problems.

Section 4 precisely state the hypothesis of polynomial-time symmetry of information as well as some simple results about this. Finally, Section 5 proves the main result, namely that polynomial-time symmetry of information implies that exponential-time problems do not have polynomial-size circuits.

## 2 Preliminaries

For references on computational complexity we recommend the book [2]. For Kolmogorov complexity, we refer to [8]. The notions used in this paper are standard, though stated from the unifying point of view of universal Turing machines.

*Universal machines.* If $M$ is a Turing machine, for simplicity we will assume that it is encoded in binary and denote by $M$ this encoding. Therefore $M$ is also seen as a program. We restrict ourselves to two-tape Turing machines; the following result on a universal Turing machine is then well-known.

**Proposition 1.** *There exists a universal Turing machine $\mathcal{U}$, with two tapes, which, on input $(M, x)$, simulates the two-tape machine $M$ on input $x$. There is a constant $c > 0$ depending only on the machine $M$ such that the simulation of $t$ steps of $M(x)$ takes $ct$ steps of $\mathcal{U}$.*

Such a universal Turing machine $\mathcal{U}$ is fixed in the remainder of the paper. The machine $M$ simulated by $\mathcal{U}$ will also be called the *program* of $\mathcal{U}$. For instance, we will say that the program $M$ decides the language $A$ if for all $x$, the computation $\mathcal{U}(M, x)$ halts, and it accepts iff $x \in A$.

*Complexity classes.* If $t : \mathbf{N} \to \mathbf{N}$ is a function, the class $\mathsf{DTIME}(t(n))$ is the set of languages $A$ recognized in time $O(t(n))$. More precisely, $A \in \mathsf{DTIME}(t(n))$ if there exist a constant $c > 0$ and a fixed program $M \in \{0,1\}^*$ such that for all word $x$, the computation $\mathcal{U}(M, x)$ stops before $ct(|x|)$ steps and it accepts if and only if $x \in A$. We call $\mathsf{EXP}$ the class $\cup_{k \geq 0} \mathsf{DTIME}(2^{n^k})$.

Now, nonuniform computation is defined via advices as introduced by Karp and Lipton [5]. The advice class $\mathsf{DTIME}(t(n))/a(n)$ is the set of languages $A$ such that there exist a program $M$, a constant $c > 0$ and a family $(a_n)$ of advices (that is to say, words) satisfying:

1. $|a_n| \leq a(n)$;
2. $\mathcal{U}(M, x, a_{|x|})$ stops in less than $ct(|x| + a(|x|))$ steps;
3. $\mathcal{U}(M, x, a_{|x|})$ accepts iff $x \in A$.

The nonuniform class $\mathsf{P/poly}$ is defined as $\cup_{k \geq 0} \mathsf{DTIME}(n^k)/n^k$ (i.e. polynomial working time and polynomial-size advice) and is easily shown to be the set of languages recognized by a family of polynomial-size boolean circuits. Similarly, $\mathsf{EXP/poly}$ is the class $\cup_{k \geq 0} \mathsf{DTIME}(2^{n^k})/n^k$ (i.e. exponential working time and polynomial-size advice). By this definition, it is easy to see that

$$\mathsf{EXP} \subset \mathsf{P/poly} \iff \mathsf{EXP/poly} = \mathsf{P/poly}.$$

Another complexity measure is the space needed to decide a language. Space complexity counts the number of cells used by the machine. The class $\mathsf{DSPACE}(s(n))$ is the set of languages $A$ recognized in space $O(s(n))$, and advice classes are defined accordingly. The class $\mathsf{PSPACE}$ is $\cup_{k\geq 0}\mathsf{DSPACE}(n^k)$.

In this paper, we shall also quickly meet the complexity class $\mathsf{PP}$. This is the set of languages $A$ such that there exist a language $B \in \mathsf{P}$ and a polynomial $p(n)$ satisfying $x \in A \iff \#\{y \in \{0,1\}^{p(|x|)} \ : \ (x,y) \in B\} \geq 2^{p(|x|)-1}$.

*Resource-bounded Kolmogorov complexity.* For two words $x, y$ and an integer $t$, we denote by $C^t(x|y)$ the time $t$ bounded Kolmogorov complexity of $x$ conditional to $y$, that is, the size of a shortest program $M$ which, when run on the universal Turing machine $\mathcal{U}$ on input $y$, outputs $x$ in time $\leq t$. For a Turing machine $M$ (and in particular for $\mathcal{U}$), we denote by $M^t(x)$ the word written on the output tape of the machine $M$ after $t$ steps of computation on input $x$. Thus in symbols we have

$$C^t(x|y) = \min\{k \ : \ \exists M \text{ of size } k \text{ such that } \mathcal{U}^t(M,y) = x\}.$$

We will use the notation $C^t(x)$ for $C^t(x|\epsilon)$, where $\epsilon$ is the empty word.

*Advice and programs.* For a fixed word length $n$, the words $x \in \{0,1\}^n$ of size $n$ are lexicographically ordered and the $i$-th one is called $x^{(i)}$ (for $1 \leq i \leq 2^n$). Let $A$ be a language. The *characteristic string* of $A^{=n}$ is the word $\chi \in \{0,1\}^{2^n}$ defined by $\chi_i = 1$ iff $x^{(i)} \in A$. We will often consider programs that output characteristic strings rather than programs that decide languages. We rely on the following obvious lemma.

**Lemma 1.** *If $A$ is a language in $\mathsf{DTIME}(t(n))/a(n)$ (where $t(n) \geq n$), then there exist constants $\alpha, k > 0$ and a family $(M_n)$ of programs satisfying:*

1. *$|M_n| \leq k + a(n)$;*
2. *for $1 \leq i \leq 2^n$ in binary, $\mathcal{U}(M_n, i)$ outputs the $i$ first bits of the characteristic string $\chi$ of $A^{=n}$ in time $\alpha i t(n + a(n))$.*

*Proof.* Let $M$ be a $\mathsf{DTIME}(t(n))$ machine deciding $A$ with advice of size $a(n)$. The program $M_n$ merely enumerates the $i$ first words $x$ of size $n$ and simulates $M(x)$: $M_n$ is therefore composed of the code of $M$, of an enumeration routine for the $i$ first words of size $n$ and of the advice for the length $n$. □

## 3 Diagonalizing out of $n^c$ advice length

We provide another proof of the following proposition of Homer and Mocas [3]. The initial proof of [3] makes use of resource-bounded Kolmogorov complexity. Here it consists in a usual diagonalization, similar to the proof of Schöning [12] that EXPSPACE does not have polynomial-size circuits (see [2, Th. 5.6]): at each step of the diagonalization process, we eliminate half of the possible programs. This easy proof can be considered as folklore; still we include it in the

present paper since it seems unpublished so far, and because it introduces some techniques used in the proof of the main result of this paper. Furthermore, this method yields a nonuniform lower bound on PP problems as a corollary. This lower bound is rather weak and a better one is already known; we include it here only as another application of the method.

**Proposition 2.** *For all constants $c_1, c_2 \geq 1$, there is a sparse language $A$ in* $\mathsf{DTIME}(2^{O(n^{1+c_1c_2})})$ *but not in* $\mathsf{DTIME}(2^{O(n^{c_1})})/n^{c_2}$.

*Proof.* The idea of the proof is to diagonalize against all programs of size $n^{c_2}$ thanks to a language that eliminates half of them at each step.

Let us define $A^{=n}$ for all $n$, therefore fix $n$. Recall that $x^{(1)} < x^{(2)} < \ldots < x^{(2^n)}$ are the words of $\{0,1\}^n$ sorted in lexicographic order. We will diagonalize over the programs $M$ of size at most $n + n^{c_2}$ (of which there are $2^{n+n^{c_2}+1} - 1$), and the universal machine $\mathcal{U}$ will be simulated for $t(n) = 2^{n^{1+c_1c_2}}$ steps. The set $A^{=n}$ is defined word by word as follows:

$$x^{(1)} \in A^{=n} \iff \begin{array}{l}\text{for at least half of the programs } M \text{ of size } \leq n + n^{c_2},\\ \text{the first bit of } \mathcal{U}^{t(n)}(M) \text{ is } 0,\end{array}$$

that is, at least half of the programs give the wrong answer for $x^{(1)}$. Let $V_1$ be the set of programs $M$ giving the right answer for $x^{(1)}$, i.e. such that the first bit of $\mathcal{U}^{t(n)}(M)$ corresponds to "$x^{(1)} \in A$". Hence $|V_1| < 2^{n+n^{c_2}}$ (less than half of the programs of size $\leq n + n^{c_2}$ remain). We then go on with $x^{(2)}$:

$$x^{(2)} \in A^{=n} \iff \begin{array}{l}\text{for at least half of the programs } M \in V_1,\\ \text{the second bit of } \mathcal{U}^{t(n)}(M) \text{ is } 0,\end{array}$$

that is, among the programs that were right for $x^{(1)}$, at least half make a mistake for $x^{(2)}$. Let $V_2$ be the set of programs $M \in V_1$ giving the right answer for $x^{(2)}$. We go on like this:

$$x^{(i)} \in A^{=n} \iff \begin{array}{l}\text{for at least half of the programs } M \in V_{i-1},\\ \text{the } i\text{-th bit of } \mathcal{U}^{t(n)}(M) \text{ is } 0\end{array}$$

until $V_i$ is empty. Call $k$ the first $i$ such that $V_i = \emptyset$. We decide arbitrarily that $x^{(j)} \notin A^{=n}$ for $j > k$. Note that $k \leq n + n^{c_2} + 1$ because $|V_i|$ is halved at each step, therefore $A$ is sparse.

If $A \in \mathsf{DTIME}(2^{O(n^{c_1})})/n^{c_2}$, then by Lemma 1 there would be a constant $k$ and a family $(M_n)$ of programs of size $\leq k + n^{c_2}$ writing down the characteristic string of $A^{=n}$ in time $\alpha(n+n^{c_2}+1)2^{O(n^{c_1c_2})} \leq 2^{\beta n^{c_1c_2}}$ for some $\beta$. This is not possible as soon as $n \geq k$ and $t(n) > 2^{\beta n^{c_1c_2}}$ since all programs of size $n + n^{c_2}$ must make a mistake on some input of size $n$. Therefore $A \notin \mathsf{DTIME}(2^{O(n^{c_1})})/n^{c_2}$.

Now, in order to decide if $x^{(i)} \in A$ it is enough to decide if $x^{(j)} \in A$ for all $j \leq i$. This is done in the order $j = 1, \ldots, i$ because we need the answer of $j$ for $j + 1$. For $x^{(j)}$ we proceed as follows: we enumerate all the programs $M$ of size $\leq n + n^{c_2}$, compute $\mathcal{U}^{t(n)}(M)$ by simulating $\mathcal{U}$ for $t(n)$ steps, we test

whether $M \in V_{j-1}$ (this is done by comparing for each $k < j$ the $k$-th bit of $\mathcal{U}^{t(n)}(M)$ with the already computed value of "$x^{(k)} \in A$"), and count how many $M \in V_{j-1}$ produce an output whose $j$-th bit is 0. If there are more than half such $M$, then $x^{(j)} \in A$, otherwise $x^{(j)} \notin A$. The overall running time of this algorithm is $(n + n^{c_2})2^{O(n^{c_2})}t(n)$, thus $A \in \mathsf{DTIME}(2^{O(n^{1+c_1c_2})})$. □

The same proof also works for space complexity.

**Proposition 3.** *For all constants $c_1, c_2 \geq 1$, there is a sparse language $A$ in $\mathsf{DSPACE}(n^{1+c_1c_2})$ but not in $\mathsf{DSPACE}(n^{c_1})/n^{c_2}$.*

The following corollary is now immediate.

**Corollary 1.** *For every constant $c > 0$, $\mathsf{EXP} \not\subset \left(\mathsf{P}/n^c\right)$ and $\mathsf{PSPACE} \not\subset (\cup_k \mathsf{DSPACE}(\log^k n)/n^c)$.*

*Remark 1.* The original result of Homer and Mocas [3] has since been improved by Ronneburger [11, Th. 5.21] in the following way: there is a language $R \in \mathsf{EXP}$ such that for all $k$, there exists a language $L \in \mathsf{EXP}$ which is not truth-table reducible to $R$ in time $2^{n^k}$ with $n^k$ bits of advice.

The construction of the language $A$ of Proposition 2 enables us to prove the following nonuniform lower bound for PP problems. As already mentioned, this is a much weaker result than Vinodchandran [13] showing that PP does not have circuits of size $n^k$ for any fixed $k$.

**Proposition 4.** *For any fixed $k > 0$, $\mathsf{PP} \not\subset \mathsf{DTIME}(n^k)/(n - \log n)$.*

*Proof.* The idea relies on the following remark: in the proof of Proposition 2, if the simulation time of the machine $\mathcal{U}$ is polynomial, then deciding whether "for at least half of the programs $M \in V_{i-1}$, the $i$-th bit of $\mathcal{U}^{t(n)}(M)$ is 0" is a PP problem.

Let us now fill the details. Take $t(n) = n^{3+k}$ for the simulation time and diagonalize over programs of size $\leq n - (\log n)/2$ (of which there are less than $2^n/\sqrt{n}$). For conveniency, if $k \leq n+1-(\log n)/2$ and $b_1, \ldots, b_k$ are $k$ bits, define

$$B(b_1, \ldots, b_k) = \{M |\ M \text{ is a program of size } \leq n - (\log n)/2 \text{ such that } \forall i \leq k, \text{ the } i\text{-th bit of } \mathcal{U}^{t(n)}(M) \text{ is } b_i\}.$$

Let us now define the following language $C$:

$$C = \{(b_1 \ldots b_{k+1}, 0^{m_k}) |\ \text{for at least half of the programs } M \in B(b_1, \ldots, b_k), \text{ the } (k+1)\text{-th bit of } \mathcal{U}^{t(n)}(M) \text{ is } b_{k+1}\}.$$

The second term $0^{m_k}$ of the couple in $C$ is only a padding term, so that the length of the queries to $C$ will be always the same. Since $k \leq n+1-(\log n)/2$, one can assume by choosing an appropriate encoding that the length of $(b_1 \ldots b_{k+1}, 0^{m_k})$ is always $n$.

Note that deciding whether $M \in B(b_1, \ldots, b_k)$ can be done in polynomial time (there are $k \leq n + 1 - (\log n)/2$ simulations to perform, each of which requires time $O(t(n))$). Therefore $C \in \mathsf{PP}$.

We can now define our main language $A$ very similarly as in Proposition 2:

$$x^{(1)} \in A^{=n} \iff (0, 0^{m_1}) \in C,$$

that is, $x^{(1)} \in A$ if and only if at least half of the programs of size $\leq n-(\log n)/2$ reject $x^{(1)}$ (i.e., they make a mistake on $x^{(1)}$). Call $b_1 \in \{0,1\}$ the right answer for $x^{(1)}$, that is, $b_1 = 1$ iff $x^{(1)} \in A^{=n}$. Then we go on with $x^{(2)}$:

$$x^{(2)} \in A^{=n} \iff (b_1 0, 0^{m_2}) \in C.$$

Once again, among the programs that were right for $x^{(1)}$, at least one half give the wrong answer for $x^{(2)}$. Going on like this for $n + 1 - (\log n)/2$ steps, all the programs are wrong on at least one word because every program has been diagonalized against. We decide arbitrarily that $x^{(i)} \notin A^{=n}$ for $i > n + 1 - (\log n)/2$. We thus have $A \notin \mathsf{DTIME}(n^{k+2})/(n - \log n)$.

The language $A$ can then be recognized in time $O(n^2)$ with oracle access to $C$: it is enough to decide successively whether $x^{(1)} \in A$, $x^{(2)} \in A$, etc. These $n + 1 - (\log n)/2$ steps can be done thanks $n + 1 - (\log n)/2$ queries of size $n$ to $C$ (the time complexity of the algorithm is $O(n^2)$ because it has to ask $O(n)$ questions of size $O(n)$).

Suppose for a contradiction that $\mathsf{PP} \subset \mathsf{DTIME}(n^k)/(n - \log n)$. Then $C \in \mathsf{DTIME}(n^k)/(n - \log n)$ and the algorithm above only queries words of size $n$. Hence $A \in \mathsf{DTIME}(n^{k+2})/(n - \log n)$ which is a contradiction. □

## 4 Symmetry of information

In this section we state the hypothesis of resource-bounded symmetry of information we will use. For the sake of completeness, we first state a version of symmetry of information for exponential time bounds. For a proof one can easily adapt the unbounded case, see for instance [8, Th. 2.8.2 p. 182].

**Theorem 1.** *There exist constants $\alpha, \beta$ such that for all words $x, y$ and all time bound $t$, the following equality holds:*

$$C^t(x, y) \geq C^{t2^{\alpha(|x|+|y|)}}(x) + C^{t2^{\alpha(|x|+|y|)}}(y|x) - \beta \log(|x| + |y|).$$

Notice that the other inequality also holds up to a logarithmic factor and is much easier to show. Here we are only interested in the "hard part" of symmetry of information. This is an open question whether this inequality holds for polynomial time bounds, i.e. whether there exists a polynomial $q(n)$ such that $C^t(x, y) \geq C^{tq(|x|+|y|)}(x) + C^{tq(|x|+|y|)}(y|x) - \beta \log(|x|+|y|)$. However, if one-way functions exist (as is often believed), then polynomial-time symmetry of information does not hold, see [9]. This suggests that this version of polynomial-time symmetry is too strong.

One way to relax this hypothesis is to allow a larger error, replacing the $O(\log(|x|+|y|))$ error term by $\delta(|x|+|y|)$. This would imply that polynomial-time computable functions can be inverted in time $2^{O(\delta(n))}$. This does not seem completely impossible if $\delta(n)$ is large enough, for instance $\delta(n) = \epsilon n$. Note also that the hypothesis is true for $\delta(n) = n$, since trivially there exists a polynomial $q$ such that $2C^t(x,y) \geq C^{tq(|x|+|y|)}(x) + C^{tq(|x|+|y|)}(y|x)$. All these remarks lead us to the following version of the hypothesis of polynomial-time symmetry of information. It is interesting to note that Lee and Romashchenko [7], in the introduction of their paper, already ask a very similar question: can we show that $(2-\epsilon)C(x,y) \geq C(x) + C(y|x)$ for some contant $\epsilon > 0$ when polynomial-time bounds are required?

---

(SI) There exist a constant $\alpha > 1/2$ and a polynomial $q$ such that for all time bound $t$ and all words $x, y, z$ of size $|x|+|y|+|z| = n$,

$$C^t(x,y|z) \geq \alpha\Big(C^{tq(n)}(x|z) + C^{tq(n)}(y|x,z)\Big).$$

---

Note that we need time bounds $tq(n)$ in the right-hand side, instead of $q(t)$ in the usual settings of polynomial-time symmetry of information, in order to limit the growth of the time bound when iteratively applying (SI). It would be nice to rule this problem out and use the usual $q(t)$ time bound instead. Note finally that the hypothesis can be weakened, as mentioned in the following remark.

*Remark 2.* For our purpose, the following restrictions can further be applied on the hypothesis (SI):

1. we can require $|x| = |y|$ and $C^{tq(n)}(x|z) = C^{tq(n)}(y|x,z)$;
2. the time bound $t$ can be taken $< 2^{n^2}$ and in this framework, the hypothesis can hold only for all but a finite number of words $x$ and $y$;
3. the constant $\alpha$ can be replaced by the nonconstant term $1/2 + 1/\sqrt{\log(|x|+|y|)}$, which is closer to $1/2$.
4. the multiplicative time bound $q(n)$ can in fact be much larger than a polynomial: we could take $q(n) = 2^{2^{\sqrt{\log n}}}$, which is greater than $2^{\log^k n}$ for all $k$. All we need is a function $q$ such that $q(n^{\log n})^{\log^2 n} < 2^{n^k}$ for some $k$.

Putting these points together, here is the weaker (but more complicated) hypothesis we obtain:
Let $f(n) = 2^{2^{\sqrt{\log n}}}$. For all $n$, all $t < f(n)$, all word $z$ and all but finitely many words $x, y$ of same length, if $|x|+|y|+|z| = n$ and $C^{tq(n)}(x|z) = C^{tq(n)}(y|x,z)$, then

$$C^t(x,y|z) \geq \left(\frac{1}{2} + \frac{1}{\sqrt{\log(|x|+|y|)}}\right)\Big(C^{tf(n)}(x|z) + C^{tf(n)}(y|x,z)\Big).$$

Note that relative to the oracle $O = \{(M,x,b) : M(x) = b \text{ in } \leq 2^{|x|} \text{ steps}\}$, (SI) is true. Furthermore, since our proofs below relativize and the conclusion does not, we obtain the following proposition (certainly also provable directly).

**Proposition 5.** *There exists an oracle $A$ relative to which* (SI) *is true and an oracle $B$ relative to which* (SI) *is false.*

We wish to iteratively apply (SI). In order to do that, we need to precise how we encode tuples. The main property we will need is that $(x_1, \ldots, x_{2n})$ should have the same encoding as $((x_1, \ldots, x_n), (x_{n+1}, \ldots, x_{2n}))$. This is achieved for instance by representing three symbols "zero", "one" and a delimiter # by 00, 11 and 01. Hence the size of the encoding of an $n$-tuple $(x_1, \ldots, x_n)$ will be $2(n-1) + 2(|x_1| + \ldots + |x_n|)$. Of course, much shorter encodings could also be chosen, but it would not help in this paper.

**Lemma 2.** *Suppose* (SI) *holds and take a corresponding polynomial $q$. Let $t$ be a time bound, $u_1, \ldots, u_n$ be words of size $s$ and $z$ be another word of arbitrary size. We define $m = ns + |z|$ the size of all these words. Suppose there exists a constant $k$ such that for all $j \leq n$, we have $C^{q(m)^{\log n} t}(u_j | u_1, \ldots, u_{j-1}, z) \geq k$.*

*Then $C^t(u_1, \ldots, u_n | z) \geq (2\alpha)^{\lfloor \log n \rfloor} k$.*

*Proof.* Fix a sequence of words $(u_i)_{i \geq 1}$ of size $s$. Let us first show the result when $n$ is a power of 2. We show by induction on $n$ (only for powers of 2) the following hypothesis: for every time bound $t$, every word $z$ and all $m \geq ns + |z|$, if for all $j \leq n$, $C^{q(m)^{\log n} t}(u_j | u_1, \ldots, u_{j-1}, z) \geq k$ then $C^t(u_1, \ldots, u_n | z) \geq (2\alpha)^{\log n} k$.

This is clear for $n = 1$. For $n > 1$, take $t, z$ and $m \geq ns + |z|$. By (SI),

$$C^t(u_1, \ldots, u_n | z) \geq \alpha \Big( C^{tq(m)}(u_1, \ldots, u_{n/2} | z) + C^{tq(m)}(u_{n/2+1}, \ldots, u_n | u_1, \ldots, u_{n/2}, z) \Big).$$

By induction hypothesis at rank $n/2$, for the time bound $tq(m)$ and where for the last term we take as new $z$ the word $u_1, \ldots, u_{n/2}, z$, the right-hand side is at least $\alpha((2\alpha)^{\log(n/2)} k + (2\alpha)^{\log(n/2)} k) = (2\alpha)^{\log n} k$.

Now, if $n$ is not a power of 2, let $p$ be the largest power of 2 less than $n$. Then $C^t(u_1, \ldots, u_n | z) \geq C^t(u_1, \ldots, u_p | z) \geq (2\alpha)^{\log p} k = (2\alpha)^{\lfloor \log n \rfloor} k$. □

We now establish links between the Kolmogorov complexity of a characteristic string and the length of the advice.

**Lemma 3.** *Let $A$ be a language and $\chi^{(n)}$ the characteristic string of $A^{=n}$ (i.e. $\chi^{(n)}_i = 1$ iff $x^{(i)} \in A^{=n}$). We denote by $\chi^{(n)}[1..i]$ the string consisting of the $i$ first bits of $\chi^{(n)}$. Let $r(n)$ be a function and suppose that there exists an unbounded function $s(n) \geq 0$ such that for all constant $\alpha > 0$, there exist infinitely many $n$ and $1 \leq i \leq 2^n$ satisfying $C^{\alpha i r(n + a(n))}(\chi^{(n)}[1..i]) > s(n) + a(n)$.*

*Then $A \notin \mathsf{DTIME}(r(n))/a(n)$.*

*Proof.* Suppose that $A \in \mathsf{DTIME}(r(n))/a(n)$. Then there exist a fixed program $M$ together with an advice function $c(n)$ of size $\leq a(n)$, such that for all $x \in \{0,1\}^n$, $\mathcal{U}(M, x, c(n))$ works in time $O(r(n + a(n)))$ and accepts iff $x \in A^{=n}$. By

enumerating the first $i$ words of size $n$ in lexicographic order and simulating the program $M$ on each of them, there is another program $N$ with advice $c(n)$ that enumerates $\chi^{(n)}[1..i]$ in time $O(ir(n+a(n)))$. Hence there exists a constant $\alpha > 0$ such that for all $n$ and for all $i \leq 2^n$, $C^{\alpha ir(n+a(n))}(\chi^{(n)}[1..i]) \leq |N| + a(n)$. □

## 5 Diagonalizing out of polynomial advice length

We are now interested in diagonalizing over all polynomial advices, not just of size $n^c$ for some fixed $c$. We will use the hypothesis of symmetry of information above. Here the main difficulty is to diagonalize over all advices without enumerating them, otherwise we would go outside of EXP. The idea is simple:

- Two different "useful" and "independent" parts of size $k_1$ and $k_2$ of an advice "must" carry roughly $k_1 + k_2$ bits of information.
- We therefore decompose the advice in small blocks (of size $O(n)$) and diagonalize over the blocks instead of the whole advice, while making sure that these blocks are "independent". The hypothesis (SI) then enables us to "glue" these blocks together.

This can also be seen as efficiently finding a string of high Kolmogorov complexity and Lemma 2 helps us do that.

**Theorem 2.** *If* (SI) *holds true, then* $\mathsf{EXP} \not\subset \mathsf{P/poly}$.

*Proof.* Suppose (SI) holds true: this gives a corresponding polynomial $q$. We diagonalize over programs ("blocks") $M$ of length $\leq n-1$ and simulate the universal machine $\mathcal{U}$ for $t(n) = q(n^{1+\log n})^{\log^2 n} n^{2+\log n+\log^3 n}$ steps. Define the language $A$ as follows, by length as in the proof of Proposition 2. The $n$ first steps of the definition are the same as for Proposition 2; the difference occurs only after, when we reuse the initial segment of $A^{=n}$ in our simulation. So we define for $i \leq n$:

$$x^{(i)} \in A^{=n} \iff \begin{array}{l}\text{for at least half of the programs } M \in V_{i-1}^{(0)},\\ \text{the } i\text{-th bit of } \mathcal{U}^{t(n)}(M) \text{ is } 0,\end{array}$$

where $V_{i-1}^{(0)}$ is the set of the remaining programs of size $\leq n-1$ which give the right answer for $x^{(i-1)}$. Remark that $V_n^{(0)} = \emptyset$ since all of the $2^n - 1$ programs $M$ have been eliminated.

Call $u^{(1)}$ the characteristic string of the initial segment of size $n$ of $A^{=n}$ defined above: thus $|u^{(1)}| = n$ and for $i \leq n$, $u_i^{(1)} = 1$ if and only if $x^{(i)} \in A^{=n}$. We now define the next segment of size $n$ of $A^{=n}$.

$$x^{(n+1)} \in A^{=n} \iff \begin{array}{l}\text{for at least half of the programs } M \text{ of size } \leq n-1,\\ \text{the first bit of } \mathcal{U}^{t(n)}(M, u^{(1)}) \text{ is } 0,\end{array}$$

that is, at least half of the programs $M$ of length $\leq n-1$ give the wrong answer for $x^{(n+1)}$ even with the string $u^{(1)}$ as advice. Let $V_1^{(1)}$ be the set of programs

$M$ giving the right answer for $x^{(n+1)}$, i.e. such that the first bit of $\mathcal{U}^{t(n)}(M, u^{(1)})$ corresponds to "$x^{(n+1)} \in A$". Hence $|V_1^{(1)}| < 2^{n-1}$. We then go on like this:

$$x^{(n+i)} \in A^{=n} \iff \begin{array}{l}\text{for at least half of the programs } M \in V_{i-1}^{(1)},\\ \text{the } i\text{-th bit of } \mathcal{U}^{t(n)}(M, u^{(1)}) \text{ is 0.}\end{array}$$

Call $u^{(2)}$ the characteristic string of the second segment of size $n$ of $A^{=n}$ defined above: thus $|u^{(2)}| = n$ and $u_i^{(2)} = 1$ if and only if $x^{(n+i)} \in A^{=n}$. We define the third segment of size $n$ of $A^{=n}$ analogously:

$$x^{(2n+1)} \in A^{=n} \iff \begin{array}{l}\text{for at least half of the programs } M \text{ of size } \leq n-1,\\ \text{the first bit of } \mathcal{U}^{t(n)}(M, u^{(1)}, u^{(2)}) \text{ is 0,}\end{array}$$

etc. Going on like this we have (for the $(j+1)$-th segment):

$$x^{(jn+i)} \in A^{=n} \iff \begin{array}{l}\text{for at least half of the programs } M \in V_{i-1}^{(j)},\\ \text{the } i\text{-th bit of } \mathcal{U}^{t(n)}(M, u^{(1)}, u^{(2)}, \ldots, u^{(j)}) \text{ is 0.}\end{array}$$

We stop when $j = n^{\log n}$ and decide arbitrarily that $x^{(k)} \notin A$ for $k > n \times n^{\log n}$.

Let us first show that $A \notin \mathsf{P/poly}$. Note that at each step of the definition of $A$, we have $C^{t(n)}(u^{(j)}|u^{(1)} \cdots u^{(j-1)}) \geq n$ because no program of length $\leq n-1$ writes $u^{(j)}$ in time $\leq t(n)$ on input $u^{(1)} \cdots u^{(j-1)}$. Since (SI) holds and by definition of $t(n)$, Lemma 2 asserts that for $i = n \times n^{\log n}$, $C^{i.n^{1+\log^3 n}}(\chi^{(n)}[1..i]) \geq (2\alpha)^{\log^2 n} n = n^{1+\log(2\alpha)\log n}$ for large enough $n$.

Hence by Lemma 3, if we let $a(n) = n^{\log(2\alpha)\log n} - n$ and $r(n) = n^{\log n}$, we have $A \notin \mathsf{DTIME}(r(n))/a(n)$. In particular, $A \notin \mathsf{P/poly}$.

It is straightforward to see that $A \in \mathsf{EXP}$, and the theorem follows. □

*Remark 3.* The same proof also works for space complexity if we assume a corresponding version of symmetry of information for polylogarithmic space bounded Kolmogorov complexity. That is, under such an assumption we can prove that $\mathsf{PSPACE} \not\subset (\cup_k \mathsf{DSPACE}(\log^k n)/\mathsf{poly})$.

Let us now give a consequence of symmetry of information on randomized algorithms. The following theorem of Nisan and Wigderson [10] will be useful for our purpose. By approximating a problem we mean an algorithm that is right on all but a fraction $1/f(n)$ of the inputs, where $f(n)$ is superpolynomial (see [10]).

**Theorem 3.** *If there exists $\epsilon > 0$ such that* $\mathsf{EXP}$ *cannot be approximated by circuits of size $2^{n^\epsilon}$ then there exists $c > 0$ such that* $\mathsf{BPP} \subseteq \mathsf{DTIME}(2^{\log^c n})$.

We can use this theorem as follows. In the proof of Theorem 2, if we build segments of size $2^{n^\epsilon}$ (instead of $n^{1+\log n}$) for $\epsilon < 1$ and repeat the process for each segment until we fill $\{0,1\}^n$, then it is easy to see that every program of size $2^{n^\epsilon}$ must make a mistake not only on one, but on a fraction $\geq 1/(2n^\epsilon)$ of the inputs (otherwise the segments could be compressed by encoding separately the "good" program and the positions where it makes a mistake). That is, assuming (SI), $\mathsf{EXP}$ cannot be approximated by circuits of size $2^{n^\epsilon}$, in the sense of [10]. Theorem 3 therefore yields the following corollary.

**Corollary 2.** *If* (SI) *holds then there is $c > 0$ such that* $\mathsf{BPP} \subseteq \mathsf{DTIME}(2^{\log^c n})$.

## 6 Further research and acknowledgement

It would be interesting to overcome the problem with $q(t)$ time bounds in the hypothesis (SI) (instead of $tq(n)$), in order to be able to use the usual statement of polynomial-time symmetry of information. Then one could try to obtain unconditional results by using variants of resource-bounded Kolmogorov complexity such as CBP or CAMD (see [7]).

The author is really indebted to Andrei Romashchenko for the useful and numerous discussions on this paper and on Kolmogorov complexity (in particular on symmetry of information). He also wants to thank Pascal Koiran for pointing out the open problem "$\mathsf{EXP} \subset \mathsf{P/poly}$?", and anonymous referees for useful comments.

## References

1. L. M. Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th IEEE Symposium on Foundations of Computer Science*, pages 75–83, October 1978.
2. J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Number 11 in EATCS monographs on theoretical computer science. Springer, 1988.
3. S. Homer and S. Mocas. Nonuniform lower bounds for exponential time classes. In *20th symposium on Mathematical Foundations of Computer Science*, volume 969 of *Lecture Notes in Computer Science*, pages 159–168. Springer, 1995.
4. R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55:40–56, 1982.
5. R. Karp and R. Lipton. Turing machines that take advice. *L'Enseignement Mathématique*, 28:191–209, 1982.
6. A. Kolmogorov. Combinatorial foundations of information theory and the calculus of probabilities. *Russian Mathematical Surveys*, 38(4):29–40, 1983.
7. T. Lee and A. Romashchenko. Resource bounded symmetry of information revisited. *Theoretical Computer Science*, 345(2–3):386–405, 2005. Earlier version in 29th Symposium on the Mathematical Foundations of Computer Science, 2004.
8. M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications.* Graduate texts in computer science. Springer, second edition, 1997.
9. L. Longpré and O. Watanabe. On symmetry of information and polynomial time invertibility. *Information and Computation*, 121(1):14–22, August 1995.
10. N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.
11. D. Ronneburger. *Kolmogorov Complexity and Derandomization.* PhD thesis, Rutgers University, 2004.
12. U. Schöning. *Complexity and structure*, volume 211 of *Lecture Notes in Computer Science.* Springer, 1985.
13. N. V. Vinodchandran. A note on the circuit complexity of PP. In *Electronic Colloquium on Computational Complexity, Report No. 56*, July 2004.
14. A. Zvonkin and L. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorihms. *Russian Mathematical Surveys*, 25(6):83–124, 1970.